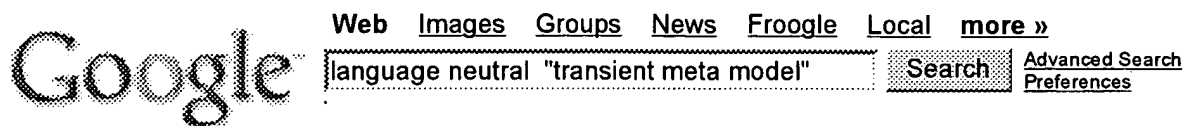


| Ref # | Hits | Search Query   | DBs   | Default Operator | Plurals | Time Stamp       |
|-------|------|--|---|------------------|---------|------------------|
| S24   | 9    | simultaneous\$2 near2 updat\$4<br>near2 (text or graphic\$4)<br><i>Rev all</i> | US-PGPUB;<br>USPAT;<br>EPO; JPO;<br>DERWENT;<br>IBM_TDB | OR               | OFF     | 2005/06/15 13:21 |
| S23   | 12   | SCI same language\$neutral   | US-PGPUB;<br>USPAT;<br>EPO; JPO;<br>DERWENT;<br>IBM_TDB | OR               | OFF     | 2005/06/15 13:18 |
| S22   | 27   | round\$trip adj engineering  | US-PGPUB;<br>USPAT;<br>EPO; JPO;<br>DERWENT;<br>IBM_TDB | OR               | OFF     | 2005/06/15 13:18 |
| S21   | 17   | source adj code adj interface same<br>generat\$4                               | US-PGPUB;<br>USPAT;<br>EPO; JPO;<br>DERWENT;<br>IBM_TDB | OR               | OFF     | 2005/06/15 13:05 |
| S20   | 66   | source adj code adj interface  | US-PGPUB;<br>USPAT;<br>EPO; JPO;<br>DERWENT;<br>IBM_TDB | OR               | OFF     | 2005/06/15 13:05 |
| S1    | 17   | uml and display same code same<br>line\$2                                      | US-PGPUB;<br>USPAT;<br>EPO; JPO;<br>DERWENT;<br>IBM_TDB | OR               | OFF     | 2005/06/15 13:04 |
| S19   | 8    | simultaneous\$2 near2 updat\$4<br>near2 (text or graphic\$4)                   | US-PGPUB;<br>USPAT;<br>EPO; JPO;<br>DERWENT;<br>IBM_TDB | OR               | OFF     | 2005/01/26 15:17 |
| S17   | 24   | round\$trip adj engineering  | US-PGPUB;<br>USPAT;<br>EPO; JPO;<br>DERWENT;<br>IBM_TDB | OR               | OFF     | 2005/01/25 11:19 |

**Web**Results 1 - 2 of 2 for **language neutral "transient meta model"**. (0.20 seconds)Did you mean: language neutral "transient **metamodel**"**EP1224541**

... with the present invention, the improved software development tool generates a **transient meta model** (TMM) 200 which stores a **language-neutral** representation of ...  
swpat.ffii.org/pikta/txt/ep/1224/541/ - 73k - Supplemental Result - [Cached](#) - [Similar pages](#)

**EP1228404**

... tool 110 also includes a **transient meta model** (TMM) 114 that is generated by the software development tool to store a **language-neutral** representation of the ...  
swpat.ffii.org/pikta/txt/ep/1228/404/ - 101k - Supplemental Result - [Cached](#) - [Similar pages](#)

Did you mean to search for: language neutral "transient **metamodel**"Free! Get the Google Toolbar. [Download Now](#) - [About Toolbar](#) [Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2005 Google


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used language neutral transient meta model

Found 6,306 of 156,259

Sort results by


[Save results to a Binder](#)
[Try an Advanced Search](#)

Display results


[Search Tips](#)
[Try this search in The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [Beyond ALBE/P: Language neutral form](#)

J. W. Lewis

 March 1981 **Proceedings of the 5th international conference on Software engineering**

 Full text available: [pdf\(595.80 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

ALBE/P is a language-based CRT editor for PASCAL programs. The CRT screen serves as a window through which a programmer can view and modify a "pretty-printed" picture of any part of a PASCAL program. The ALBE/P system differs from conventional screen-oriented text editors in that the program is stored as a PASCAL parse tree and the editing operations are designed specifically for the PASCAL language. Moreover, because ALBE/P is language-based, it will not accept programs with lo ...

### 2 [Developing language neutral class libraries with the System Object Model \(SOM\)](#)

Mike Conner, Nurcan Coskun, Scott Danforth, Larry Loucks, Andy Martin, Larry Raper, Roger Sessions

 December 1992 **ACM SIGPLAN OOPS Messenger , Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum)**, Volume 4 Issue 2

 Full text available: [pdf\(360.02 KB\)](#)

 Additional Information: [full citation](#), [references](#), [index terms](#)

### 3 [Programming languages for non-numeric processing—1: TRAC, a text handling language](#)

R. W. Floyd, C. N. Mooers, L. P. Deutsch

 August 1965 **Proceedings of the 1965 20th national conference**

 Full text available: [pdf\(1.45 MB\)](#)


 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

THE TRAC SYSTEM for Text Reckoning And Compiling was developed as a software package and user language to go with the reactive typewriter. Design goals included the attainment of a concise and efficient input language, a straightforward philosophy and a high order of logical versatility. The external and internal forms of the TRAC language are the same. TRAC can accept, name, store, operate upon in any way, and emit any string of characters that can be produced on a teletypewriter keyboard. ...

### 4 [TRAC, a procedure-describing language for the reactive typewriter](#)

Calvin N. Mooers

March 1966 **Communications of the ACM**, Volume 9 Issue 3

Full text available:  [pdf\(635.56 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

A description of the TRAC (Text Reckoning And Compiling) language and processing algorithm is given. The TRAC language was developed as the basis of a software package for the reactive typewriter. In the TRAC language, one can write procedures for accepting, naming and storing any character string from the typewriter; for modifying any string in any way; for treating any string at any time as an executable procedure, or as a name, or as text; and for printing out any string. The TRAC language ...

5 Technical Papers: Applying natural language processing (NLP) based metadata extraction to automatically acquire user preferences



Woojin Paik, Sibel Yilmazel, Eric Brown, Maryjane Poulin, Stephane Dubon, Christophe Amice  
October 2001 **Proceedings of the international conference on Knowledge capture**

Full text available:  [pdf\(210.42 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes a metadata extraction technique based on natural language processing (NLP) which extracts personalized information from email communications between financial analysts and their clients. Personalized means connecting users with content in a personally meaningful way to create, grow, and retain online relationships. Personalization often results in the creation of user profiles that store individuals' preferences regarding goods or services offered by various e-commerce merch ...

**Keywords:** metadata extraction, natural language processing, user preference elicitation

6 Technical correspondence: Language integration in the common language runtime



Jennifer Hamilton  
February 2003 **ACM SIGPLAN Notices**, Volume 38 Issue 2

Full text available:  [pdf\(974.52 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

The Common Language Runtime (CLR) is language and platform-neutral, and provides the underlying infrastructure for the Microsoft .NET Framework. A key innovation in the CLR is its support for multiple programming languages, enabling programming language integration at the runtime level to a much greater degree than is currently possible.

**Keywords:** common type system, exception handling, intermediate language, language interoperability, metadata, virtual machine

7 Informatics: program language: Translating interactive computer dialogues from ideographic to alphabetic languages



Ian H. Witten  
September 1980 **Proceedings of the 8th conference on Computational linguistics**

Full text available:  [pdf\(879.60 KB\)](#) Additional Information: [full citation](#), [references](#)

8 A computational grammar of discourse-neutral prosodic phrasing in English



J. Bachenko, E. Fitzpatrick  
September 1990 **Computational Linguistics**, Volume 16 Issue 3

Full text available:  [pdf\(1.63 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)  
[Publisher Site](#)

We describe an experimental text-to-speech system that uses information about syntactic

constituency, adjacency to a verb, and constituent length to determine prosodic phrasing for synthetic speech. A central goal of our work has been to characterize "discourse neutral" phrasing, i.e. sentence-level phrasing patterns that are independent of discourse semantics. Our account builds on Bachenko et al. (1986), but differs in its treatment of clausal structure and predicate-argument relations. Result ...

### 9 Using Java reflection to automate extension language parsing

Dale Parson

December 1999 **ACM SIGPLAN Notices , Proceedings of the 2nd conference on Domain-specific languages**, Volume 35 Issue 1

Full text available:  [pdf\(1.03 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


An extension language is an interpreted programming language designed to be embedded in a domain-specific framework. The addition of domain-specific primitive operations to an embedded extension language transforms that vanilla extension language into a domain-specific language. The LUXWORKS processor simulator and debugger from Lucent uses Tcl as its extension language. After an overview of extension language embedding and LUXWORKS experience, this paper looks at using Java reflection and ...



### 10 Design management requirements for hardware description languages

Flávio R. Wagner

December 1995 **Proceedings of the conference on European design automation**

Full text available:  [pdf\(816.91 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)



### 11 Technical correspondence: Translating EXPRESS language model into C language model

Yu ChunYan, Wu Minghui, Liu Nairuo, Zhuang Yueting, Pan Yunhe

June 2003 **ACM SIGPLAN Notices**, Volume 38 Issue 6

Full text available:  [pdf\(228.61 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

EXPRESS is a powerful object-oriented data model descriptive language and independent of any platform. However, it is a kind of descriptive language rather than a programming language. This brings difficulty to implement the EXPRESS data model on a computer and it also undoubtedly brings a big problem with the STEP data model processing so that we have to translate EXPRESS language model into a certain programming language model. In this paper, an approach is presented to translate the EXPRESS I ...


**Keywords:** C language model, EXPRESS language model, STEP standard, data model translation, model translation



### 12 Manufacturing applications: Neutral information structure for manufacturing simulations: a neutral information model for simulating machine shop operations

Y. Tina Lee, Charles McLean, Guodong Shao

December 2003 **Proceedings of the 35th conference on Winter simulation: driving innovation**

Full text available:  [pdf\(348.20 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Small machine shops typically do not have the resources to develop custom simulations of their operations or data translators to import their data from other manufacturing software applications. This paper presents an overview of an information model currently under development at the National Institute of Standards and Technology (NIST) to address this problem. The model provides neutral data interfaces for integrating machine shop software applications with simulation. The information model ...



### 13 An optimizing compiler for batches of temporal logic formulas

James Ezick

July 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis**, Volume 29 Issue 4

Full text available:  [pdf\(282.48 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Model checking based on validating temporal logic formulas has proven practical and effective for numerous software engineering applications. As systems based on this approach have become more mainstream, a need has arisen to deal effectively with large batches of formulas over a common model. Presently, most systems validate formulas one at a time, with little or no interaction between validation of separate formulas. This is the case despite the fact that, for a wide range of applications, a c ...

**Keywords:** model checking, optimizing compiler, temporal logic



### 14 Natural language generation from plans

Chris Mellish, Roger Evans

December 1989 **Computational Linguistics**, Volume 15 Issue 4

Full text available:  [pdf\(1.84 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)  
[Publisher Site](#)

This paper addresses the problem of designing a system that accepts a plan structure of the sort generated by AI planning programs and produces natural language text explaining how to execute the plan. We describe a system that generates text from plans produced by the NONLIN planner (Tate 1976). The results of our system are promising, but the texts still lack much of the smoothness of human-generated text. This is partly because, although the domain of plans seems *a priori* to provide ric ...



### 15 A comparison of Ada and Java as a foundation teaching language

Benjamin M. Brosgol

September 1998 **ACM SIGAda Ada Letters**, Volume XVIII Issue 5

Full text available:  [pdf\(1.49 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Java has entered the software arena in unprecedented fashion, upstaging languages and technologies that are longstanding players in the industry. Almost unheard of before 1995, the language and its surrounding technology are attracting increasing attention not merely in the hardware and software communities but also among lay users and in the popular press. This phenomenon has not escaped the attention of academia, and a growing number of colleges and universities are looking at Java as a candid ...



### 16 On the completeness of object-creating database transformation languages

Jan Van Den Bussche, Dirk Van Gucht, Marc Andries, Marc Gyssens

March 1997 **Journal of the ACM (JACM)**, Volume 44 Issue 2

Full text available:  [pdf\(603.60 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Object-oriented applications of database systems require database transformations involving nonstandard functionalities such as set manipulation and object creation, that is, the introduction of new domain elements. To deal with these functionalities, Abiteboul and Kanellakis [1989] introduced the "determinate" transformations as a generalization of the standard domain-preserving transformations. The obvious extensions of complete standard database programming languages, howe ...



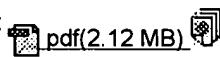
**Keywords:** computational completeness, constructive transformation, first-order logic, object creation, object-oriented database, while loop

17 Special issue on natural language generation: A generative perspective on verb alternations

Manfred Stede

September 1998 **Computational Linguistics**, Volume 24 Issue 3

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

[Publisher Site](#)

Verb alternations have been researched extensively in linguistics, but they have not yet received a systematic treatment in natural language generation systems; consequently, generators cannot make informed choices among alternatives. As a step towards overcoming this discrepancy, we review some linguistic work on several prominent alternations, revise and extend it, and suggest a set of rules that allow the series of alternated forms to be produced from a single base form of the verb, the lexic ...

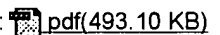


18 Sensitive parsing: error analysis and explanation in an intelligent language tutoring system

Camilla Schwind

August 1988 **Proceedings of the 12th conference on Computational linguistics - Volume 2**

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We present a uniform framework for dealing with errors in natural language sentences within the context of automated second language teaching. The idea is to use a feature grammar and to analyse errors as being sentences where features have other values than those they should have. By using a feature grammar it is possible to describe various types of errors (agreement, syntactic and semantic errors) in a uniform framework, to define in a clear and transparent way what an error is and - this is ...



19 Special issue of the lexicon: Large lexicons for natural language processing: utilising the grammar coding system of LDOCE

Bran Boguraev, Ted Briscoe

July 1987 **Computational Linguistics**, Volume 13 Issue 3-4

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

[Publisher Site](#)

This article focusses on the derivation of large lexicons for natural language processing. We describe the development of a dictionary support environment linking a restructured version of the Longman Dictionary of Contemporary English to natural language processing systems. The process of restructuring the information in the machine readable version of the dictionary is discussed. The Longman grammar code system is used to construct 'theory neutral' lexical entries. We demonstrate how such lexi ...

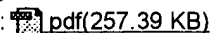


20 Student session: Aspect and discourse structure: is a *neutral* viewpoint required?

Frank Schilder

June 1995 **Proceedings of the 33rd annual meeting on Association for Computational Linguistics**

Full text available:



Additional Information: [full citation](#), [abstract](#), [references](#)

[Publisher Site](#)

We apply Smith's theory of aspect (1991) to German - a language without any aspectual markers. In particular, we try to shed more light on the effects aspect can have on



**discourse structure** and show how English and German behave differently in this respect. We furthermore describe how Smith's notion of a *neutral viewpoint* can be helpful for the analysis of discourse in German. It turned out that proposals claiming that the German *Preterite* covers the progressive as we ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.  
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

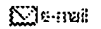


[Home](#) | [Login](#) | [Logout](#) | [Access Information](#) | [Alerts](#) |

Welcome United States Patent and Trademark Office

[Search Results](#)[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)

Results for "((language neutral 'transient meta model')&lt;in&gt;metadata)"



Your search matched 0 of 1171917 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.[» View Session History](#)[» New Search](#)[» Key](#)

Modify Search



IEEE JNL IEEE Journal or Magazine

☐ Check to search only within this results set

IEEE JNL IEE Journal or Magazine

Display Format: ☒ Citation ☐ Citation & Abstract

IEEE CNF IEEE Conference Proceeding

IEEE CNF IEE Conference Proceeding

**No results were found.**

IEEE STD IEEE Standard

Please edit your search criteria and try again. Refer to the Help pages if you need assistance revisir

[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2005 IEEE –